Metaversal Model Foundation

Overview

2024.Q3

Metaversal Model Foundation

Introduction

There are an estimated 200 million REST APIs across the internet today. These APIs resemble the antiquated hardware of the early personal computing days before device drivers made it easy for programmers to interface with them. Back then, hardware components came with proprietary communication protocols in printed manuals, and each programmer wrote virtually the exact same code to interface with them.

Today's REST APIs share similar traits. Each API is manually documented and coded, even if most solutions are virtually identical from one consumer to another. For example, if a developer wants to integrate their application with a new payment processor, they must first learn the API's protocols, and then build a solution to communicate with that system. Every other company that wants to integrate with that payment processor will inevitably build a nearly identical solution for their own application.

This problem becomes exponentially worse when you consider that the popular vision for the metaverse largely requires real-time services rather than stateless REST services. Current real-time application protocols can be arbitrarily complex and there are few, if any, standards to keep them in check. Real-time applications almost always ship with ready-made clients that are intrinsically bound in a one-to-one relationship with an application server. Due to the highly proprietary nature of today's real-time applications, scalable real-time services with persistent connections are few and far between.

The Metaversal Model Foundation (MVMF) library solves both of these problems by creating a standardized "service driver" for communicating with both stateless and real-time networked services. At the core of MVMF, client side models maintain a representation of data stored in objects on the server via common or proprietary protocols.

Unlike typical REST APIs, it is the service provider's responsibility to create, distribute, and maintain these models, freeing applications that use them from needing to understand the implementation specifics of client/service communication, and eliminating redundant programming. This will result in collective savings of trillions of hours of work for companies across every sector of computing.

What MVMF Does

MVMF is an open source, proposed open standard that provides a complete, elegant, easy-to-implement, and easier-to-consume solution to the complicated problem of interfacing with an unknown API. Service providers use MVMF to build standardized services, which can be any application, feature, or tool that shares resources over a network. Applications can effortlessly gain access to networked resources without needing to know any details about how to communicate with or receive updates from a service.

While MVMF can be used to replace any standard REST or real-time API in use today, it is specifically required to make services compatible with the spatial internet. This is because MVMF abstracts networked APIs into a common format accessible via standard classes that utilize common methods for sending and receiving data as well as signaling the readiness and modification of data.

How It Works

MVMF implements three layers of code that manage data and communication between a device (the client) and a service provider (the server). Packages published by a service provider are installed in an application. A connection to the service is made through a single line of code. Logins are supported using a simple connection via the same, or one more line of code. Once connected, gaining access to information is as easy as opening a model with yet one more line of code. Any time the status of a model changes, the application is notified with an event.

Provided that the service supplies data in an MVMF compliant format, MVMF will handle virtually all of the underlying work to keep data updated and in sync between the client and the server. If ever the service is upgraded, the service provider merely needs to update the models and publish a new package for clients. Consumers never need to know the details about how the client and server communicate. And because MVMF makes things so simple, the service provider may not need to know either.

Why It's Useful

When you design MVMF compatible services, you are inherently architecting your service in a way that is both highly reliable and highly efficient. From the service provider's perspective, MVMF makes the process of deploying and maintaining services easy. It also

provides a consistent protocol for both stateless and real-time services. From the service consumer's perspective, every model works the same way, making it simple to add new features or services to your app without having to know how each service works or how the communication channel is maintained.

One of the more powerful features of MVMF is that the built-in intermediate layers of technology make the process of keeping models up to date, even in real-time, practically effortless. Another benefit for using MVMF is that all models provide an identical interface for instantiating and destroying models, maintaining state, providing notifications when changes occur, and making requests to modify the model.

MVMF encourages the use of base classes for models that are common within an industry. For instance, base models for cars and payment processing could be standardized and maintained by their respective industries, allowing for seamless integration with any models built on those base classes.

How RP1 Uses MVMF

Each of the services that comprise RP1's spatial fabric use MVMF to manage the communication between the client and server. Furthermore, the connection strings for each service are incorporated directly into the metaverse browser standard for spatial fabrics.

Activities (dynamic objects) placed in the map rely on MVMF as well. When the service(s) for an activity are started, they run in their own memory space with an independent instance of MVMF as the foundation for communication between the metaverse browser and the service. This architecture makes it easy for millions of real-time services to seamlessly be incorporated into a shared 3D space.

Learn More

This document merely provides an introduction to and overview of the Metaversal Model Foundation. For a deeper understanding of MVMF and the role it plays in the broader scheme of the metaverse, along with additional information about metaverse browsers and spatial fabrics, download our white paper, *The Blueprint for the Open Metaverse*.

rp1.com/whitepaper

About RP1

RP1 is pioneering the next evolution of the internet with the first prototype metaverse browser that combines open standards and protocols with a proprietary software-based approach that scales to the world's population.

The metaverse browser allows 3D immersive applications to be delivered on-demand to any device, similar to how we consume web content today. RP1's spatial fabric makes it possible to navigate a 1:1 digital twin of our solar system and an entire universe from any device (PC, mobile, AR/VR) for business, education, and entertainment.

A global open demo of RP1 is coming soon at RP1.com. Contact us directly if you would like to see a demo privately.

About Metaversal Corporation

Metaversal is the world's leading technology company solving for the next generation of the spatial internet. Our mission is to build open software that connects people, services, and devices in a single, persistent metaverse.

We introduced RP1, the first prototype metaverse browser, MVMF, the first real-time API solution, and the statabase, a revolutionary network server architecture enabling unsurpassed vertical and horizontal scale. With open standards and services, RP1 will allow anyone to create and publish fully immersive XR experiences in the metaverse.

RP1.com rp1.com/discord @rp1browser